N87-26546

AN APPLICATION OF THE MPP TO THE INTERACTIVE MANIPULATION OF STEREO IMAGES OF DIGITAL TERRAIN MODELS.

Sanjay Pol, David McAllister, Edward Davis
North Carolina State University
Raleigh, North Carolina
U.S.A

ABSTRACT

We develop MPP algorithms for the interactive manipulation of flat shaded digital terrain models defined over grids. The emphasis is on real time manipulation of stereo images. Standard graphics transformations are applied to a 128 x 128 grid of elevations followed by shading and a perspective projection to produce the right eye image. The surface is then rendered using a simple painter's algorithm for hidden surface removal. The left eye image is produced by rotating the surface 6 degrees about the viewer's y axis followed by a perspective projection and rendering of the image as described above. The left and right eye images are then presented on a graphics device using standard stereo technology. Performance evaluations and comparisons are presented.

Keywords: MPP, graphics, stereo images, parallel algorithms, terrain models.

INTRODUCTION

Grid based digital terrain models contain an m x n rectangular grid of points (x_i, y_j), $1 \le i \le m$, $1 \le j \le n$, which are longitude and laltitude values on the earth's surface. Each grid point has an associated value z_{ij} which is the elevation above sea level at the point (x_i, y_j). The data is often collected by cartographers from stereo photographs of the earth's surface using a stereoplotter. The cartographer uses input devices on the stereoplotter to manipulate a virtual 3D cursor. The operator drives the cursor across the surface of the earth while the stereo plotter produces digital elevations.

Although thousands of points may be collected on a given surface, these points are massaged to produce a grid of equally spaced terrain data. In a separate operation, data are collected describing features such as rivers, lakes,bridges, etc. The cartographers then search for anomalies or inconsistencies in the two sets of data. Anomalies might include rivers which run uphill, lakes which do not sit on the surface of the earth, etc.

The Defense Mapping Agency believes that anomalies in the data can be more quickly and easily located if cartographers are able to view and interact with these digital data using a three dimensional image. Interaction with the data in real time and the ability to shade the image require a high speed computer graphics device which can produce stereo images of the data at the refresh rate of the crt or projection mechanism. The technology used to produce and view such a three dimensional image is described in [1]. The left and right eye images share the odd and even scan lines of the crt, if interlaced refresh is used, while the two images are double buffered in graphics memory and rendered alternately in the case of noninterlaced refresh. In both cases, a crt with a rapidly decaying phosphor is required to preclude ghosting. Polarization in the form of liquid crystal or plzt shutters is used to block the left eye when the right eye image is rendered and vice versa. The refresh rate must be sufficiently fast to eliminate flicker, usually at least 30 Hz interlaced or 60 Hz noninterlaced. In this paper the left and right eye images appear side by side. The stereo effect can be obtained by 'free viewing' the image. Free viewing requires the viewer to gently cross the eyes so that the two images merge to form the three dimensional image in the center. An alternative is to place a piece of paper perpendicular to the photograph so that the left eye sees only the left image and the right eye sees only the right image. Assuming a left handed coordinate system where x and y axes are centered on the viewing screen, the left eye view is formed by rotating the right eye view 6 degrees about the y axis. The two images are then drawn or rendered on the screen.

The points $P = \{(x_i, y_j, z_{ij})\}$ describe a discrete bivariate functional surface which can be rendered on a computer graphics device using standard polygon based graphics primitives. We discuss these primitives and their implementation in the next two sections. In later sections we present some examples and analyze the performace of the MPP and compare it to a VAX 11/780. Finally we offer directions for future research.

GRAPHICS PRIMITIVES

We assume m=n=128 and that the points in P are assigned to processors in the obvious way: processor p_{ij} contains the grid point (x_i, y_j, z_{ij}). Larger grids must be subdivided into 128x128 sections. The terrain surface is represented by triangles as shown in figure 1. Triangles have the advantage that they are always planar which simplifies rendering.

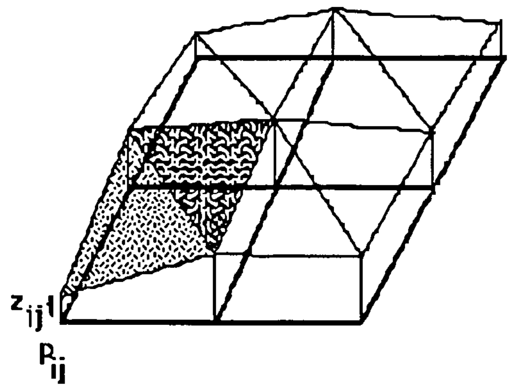

Figure 1. Terrain Model

The standard graphics operations of rotation, translation, scaling and perspective projection can be computed by matrix multiplications. Each point in P is multiplied by the transformation matrix to place the resulting terrain image in the proper position for viewing. The triangles comprising the terrain are then projected onto the viewing screen and filled with a shading intensity in an order which assures that triangles which are hidden from the viewer are eliminated. The latter process is called 'hidden surface elimination' [2].

Clipping involves presenting only that portion of the image that is visible to the user through his chosen viewing window, i.e. if the resulting image is too large to fit on the screen some of the image must be discarded before rendering. This process is computationally expensive on serial machines and we have chosen to ignore it here. We discuss it further in the final section.

The standard algorithms for hidden surface elimination are notoriously computation intensive on serial machines. They are designed to treat arbitrarily complex (and non-convex) objects which may intersect themselves, have holes, etc. However, the problem of hidden surface elimination can be solved easily for a digital terrain. The surface data can be divided into eight parts. The algorithm determines the octet number in which the chosen eye point lies based on the distances calculated to the four corners of the surface data. In any one octet the farthest point and the second farthest point are always fixed. (Figure 2).

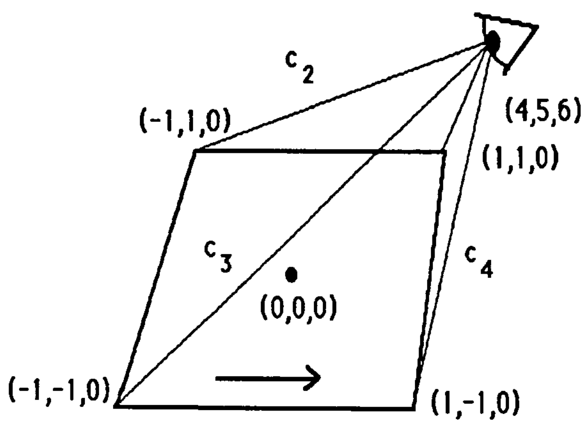

Figure 2. Rendering Order

The rendering of the terrain proceeds from the farthest to the second farthest corner. The technique is a painter's algorithm without a topological or geometric sort. For example, if the coordinates of the four corners of a rectangular grid centered at the origin are (1,1,0), (-1,1,0), (-1,-1,0), and (1,-1,0) the distances to these four corners from the eyepoint located at (4,5,6) are 25, 41, 61 and 45 respectively. The rendering therefore proceeds form corner 3 to corner 4. (Figure 2.). This technique is very effective in rendering digital terrains defined over rectangular grids. The approach is possible because a terrain can be rendered from back to front relative to the eyepoint and hidden surfaces will auomatically be removed. This algorithm makes possible real time manipulation of the stereo images because no sorting is required in the process.

With the emphasis on real time manipulation of stereo images, we have chosen to use the constant or flat shading model. In flat shading we assign the same intensity value to each pixel enclosed by a projected polygon. More sophisticated smooth shading techniques calculate separate intensities for each pixel within the polygon by interpolating the intensities across the edges of abutting polygons [3,4,5,6].

To compute the intensity or shading value to be assigned to a triangle two vectors are required: a vector of unit length perpendicular to the surface of the triangle called the unit normal and a unit vector from the centroid of the triangle to the light source (which we shall term the light vector) (Figure 3). The sun is the single light source.

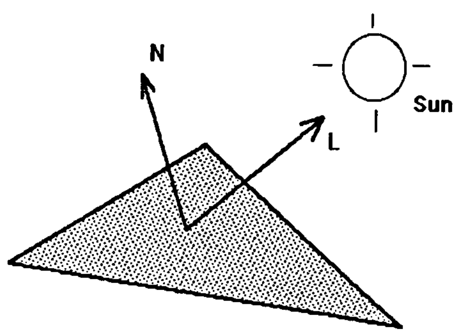

Figure 3. Flat Shading

The normal to the triangle is the cross product of any two sides of the triangle. The dot product of the normal and the light vector for the triangle is the cosine of the angle between the two and is used to determine the shade value used for illuminating the triangle. The distance to the light source is ignored. An intensity value representing ambient light is included so that all triangles are visible relative to the background color. We have not included a spectral reflection component since terrain images are not glossy.

Flat shading may result in a phenomenon called Mach banding which is the result of an anomaly in the human visual system which causes exaggeration of intensity change at any edge where there is a discontinuity in the slope of the intensity function [7]. This phenomenon is apparent in figures 4 and 5 in Color Plate I. It can be eliminated by resorting to smooth shading algorithms.

THE ALGORITHM

The algorithm which produces a stereo image is divided in two Sections. The first Section which executes on the MPP performs the necessary computations for correct orientation of the terrain surface. The second Section renders the terrain on the graphics device. All MPP related activities made use of references [8-14].

Section One contains as a subpart a preprocessing Fortran driver to produce input to a parallel Pascal routine. The Fortran driver reads or generates the digital terrain data and queries the user for the location of the eye point, and the location of the sun. The driver then constructs the necessary homogeneous transformation matrix which involves a series of matrix multiplications. The purpose of these transformations is to rotate the object so that the eyepoint lies on the Z axis. The scalar arrays for the eye point, light source, and rotation matrix are transferred to the MPP for computation of the stereo pair in Part 2 of Section One.

To compute the shade values to be assigned to each of the triangles, the normal to the surface of every triangle and the light vector are computed. These shade values are also output to the display part of the program. The total number of triangles generated for a 128 X 128 array is 32258 (127 x 127 x2). On the VAX the above computation must be done serially for each triangle. On the MPP, the computation for each triangle is assigned to one processing element. Shade values for all triangles can therefore be computed in two passes through the shading code on the MPP.

Each point in the array is now transformed in parallel by postmultiplcation with the transformation matrix. To perform the perspective transformation, each point is divided by the sum of its Z coordinate and the viewing distance. The final step in the computation is to convert the grid points to screen coordinates and to render the terrain on a graphics device.

EXAMPLES

Six stereo pairs are presented in figures 4 through 9 in the Color Plate I. All surfaces have been evaluated over a grid of 128 x 128 equally spaced points on [-1,-1] x [1,1]. In each case 256 intensity values have been used in the shading model.

Figure 4 on Color Plate I is a stereo pair of $f(x, y) = \exp[-(|x| + |y|) / 4] \cos(|x| + |y|)$. The image was chosen to demonstrate the hidden surface elimination procedure.

Mach banding is evident in this and the following figure. Figure 5 in Color Plate I is the function of figure 4 with the variable x removed. Figure 6 in Color Plate I is the usual cowboy hat function $f(x,y) = \exp(-(x^2 + y^2)) \cos(x^2 + y^2)$ while Figure 7 in Color Plate I is the function $f(x,y) = \exp(-|xy|) \cos(x^2 + y^2)$. Both surfaces appear to be smooth shaded.

Figures 8 and 9 in Color Plate I are digital terrain models representing a region in Seattle, Washington. The elevation values have been scaled up to exaggerate the depth in the image.

PERFORMANCE

The above terrain rendering algorithm has been implemented on the MPP at Goddard and on a VAX 11/780 at North Carolina State University. In one test case to compute the altitudes, perform the transformations and calculate the shade values for one of the two images the VAX used 105 seconds of CPU time. To display the data points on half of the screen on a Lexidata 90 graphics system in high resolution (1024 x 1280) requires 150 seconds of CPU time. This is the time to send the screen coordinates of the triangle vertices that form the image, their shade values and to render the image on the screen. We were not able to obtain performace statistics on the MPP because of a hardware error (failure to open Virtual Channel). However, performance estimates based on a 100 nanosecond time step for each processor, yielding a conservative overall execution rate of 200 Mflops for the PE's, shows that the computation of the transformations will require 0.02 seconds. Hence, computational aspects of real time stereo images indicate that it is possible to compute 50 stereo images per second for a 128 x 128 array on the MPP. The computation must be supported by appropriate high bandwidth I/O hardware and a fast rendering graphics device. The bandwidth required for rendering at a 60 frame per second rate using floating point data, is approximately 160 megabytes per second. Although the MPP-VAX system I/O rate is less than this, the MPP array interface is capable of this rate.

Computation rates and I/O rates can be reduced by using fixed point data with word lengths shorter than 32 bits. This is reasonable for the present problem because terrain data values can be represented in fixed point format. The highest peak on the surface of the earth is less than 32000 feet high. One foot resolution is usually sufficent which means that a 16 bit integer can be used for the elevation (z_{ij}) values. For a 128 X 128 array the longitude and latitude values can be represented as 8 bit integers. The MPP allows for these different data representations to exploit its computation, storage and I/O capabilities efficiently.

CONCLUSIONS AND FURTHER RESEARCH

Our initial studies show that grid based digital terrain models can be mapped in a natural way to the SIMD architecture of the MPP. The parallelism of the machine can be exploited in computing the necessary graphics transformations required to render stereo images of a terrain in real time on a high speed graphics device .

Development of the algorithm was inhibited by the memory limitations of the Processing Elements. The algorithm had to be divided into smaller segments to avoid overwriting on the memory bit planes used by the MPP. Additional memory will be required to handle larger grids.

Rapid identification of data anomalies may require closer inspection of parts of the terrain surface and hence zooming or scaling up of the image will be necessary. This may require clipping so the image fits on the viewing screen. Image enlargement may also require a pixel oriented smooth shading algorithm [3]. Shadowing can enhance visual realism and also aid in locating data anomalies. However, most shadowing algorithms on sequential machines are notoriously slow. Algorithms have been proposed which are adaptations of hidden surface techniques [15]. We intend to investigate these and other algorithms for possible implementation on the MPP.

REFERENCES

1. L.F.Hodges and D.F.McAllister, "Stereo and Alternating Pair Techniques for Display of Computer Generated Images", IEEE CG&A, Vol. 5, No.9, Sept.1985, pp 38-45.

2. I.E.Sutherland, R.F.Sproull and R.A.Schumacker,"A characterization of Ten Hidden-Surface Algorithms", Computing Surveys, Vol. 6, No.1, March 1974, pp. 1-55.

3. H. Gouraud, "Continuous Shading of Curved Surfaces",IEEE Transactions on Computers, Vol.C-20, No.6, June 1971, pp. 623-628.

4. Phong Bui-Tong,"Illumination for Computer Generated Pictures", Communications of the ACM, Vol.18, No. 6, June 1975, pp. 311-317.

5. K.E.Torrance and E.M.Sparrow, "Polarization, Direction Distribution, and Off-Specular Peak Phenomena in Light Reflected from Roughened Surfaces", J. Optical Society of America, vol. 56, No. 7, July 1966, pp. 916-925.

6. K.E. Torrance and E.M. Sparrow, "Theory for Off Specular Reflection from Roughened Surfaces", J. Optical Society of America, vol. 57, No. 9, Sept. 1967, pp.1105-1114.

7. F. Ratliff, Mach Bands: Quantitative Studies on Neural Networks in the Retina,Holden-Day, San Francisco, 1965.

8. General Description of the MPP, Goodyear Aerospace Corporation, GER 17140, April 1983.

9. MPP User's Guide, Edited by Jim Fischer, Science Applications Research,Lantham,MD., Jan. 1986.

10. Introduction to the MPP Host Run-Time System (MPPHRTS), Version 3.01, January 1986.

11. Parallel Pascal Language Reference Manual,MUD210, NASA, Goddard Space Flight Center.

12. Parallel Pascal User's Guide, MUD220, NASA, Goddard Space Flight Center.

13. MPP Pascal Callable Procedure Library, J. Abeles, Science Applications Research, Lantham, MD., January 1986.

14. Control and Debug (CAD) User's Manual, Goodyear Aerospace April 1983.

15. J.D.Foley & A.VanDam, Fundamentals of Interactive Computer Graphics, Addison Wesley, July 1984.